

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

**Dynamic Scalable Multi-Media Content Streaming**

Express Mail mailing label number EL 414969135 US

Date of Deposit SEPTEMBER 18, 1999

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231

Judith A. Kromyko 9-18-99  
Signature Date

Inventor(s): **Jeremy A. Kenyon**  
**Alex K. St. John**

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN, LLP  
12400 Wilshire Boulevard, 7th Floor  
Los Angeles, California 90025  
(503) 684-6200

"Express Mail" label number EL 414969135US

## **Dynamic Scalable Multi-Media Content Streaming**

### **BACKGROUND OF THE INVENTION**

5     1.     **Field of the Invention**

The present invention relates to the fields of computer systems and multi-media contents. More specifically, the present invention relates to streaming of multi-media contents.

10    2.     **Background Information**

Numerous data compression teachings are known in the art, however only a limited number of these techniques have been applied to multi-media content streaming, including but not limited to video conferencing, computerized gaming and the like. In the area of computerized games, traditionally, they were either simple single-user simulators for pinball, cards, gambling, fighting, etc., or moderately complex multiple-player turn-taking games where each player competed against the game and ultimately compared scores. Well-known high-tech examples of turn-taking games include the Nintendo® and Sony PlayStation® gaming systems; these games provide rich game content through dedicated graphics systems and local storage of graphics data. A significant drawback to these games is the inability to play against remote players and reference a fixed medium such as a CD-ROM distribution.

To overcome the remote player limitation, some games were designed to support modem communication, allowing for a game against a remote player, with the bulk of execution being performed on a remote server. However, this solution

was often clumsy, slow, and did not always work consistently. These limitations were partially overcome with the advent of widely available networking systems. Players would initiate separate modem connections to a network hub (e.g., an Internet Service Provider), which in turn provided network connections for each  
5 player. This allowed for real-time multiple-player games. However, the slow download issue over a network connection remained substantially unresolved.

The shift to multiple-player games meant that rich game content could no longer be accessed through fast retrieval from local storage, such as over a local bus to a hard disk, game cartridge, or other permanent storage. That is, computer  
10 graphics programs use geometry to describe two and three-dimensional objects within their models. In particular, complex object surfaces are usually represented by a combination of one or more basic object shapes, such as splines, non-uniform rational splines (NURBs), texture maps, and (monohedral) triangle tessellation. Typically, an arbitrary object is defined by triangle tessellation, each triangle having  
15 associated spatial coordinate tuples X, Y (and perhaps Z), color, normal, and other attributes. This information, when multiplied by hundreds or thousands of polygons in moderately complex objects, amounts to a significant bandwidth requirement for transmitting, receiving, and processing of such model data.

Current dial-up networking connections are not up to the task of transferring  
20 such amounts of model information in reasonable time. Modem connections generally realize an average modem bit rate of 14-40 KBits per second (sometimes better), and this is insufficient to directly transfer model data that has previously been retrieved from local storage context. This problem is exacerbated with each additional player, since this same data needs to be distributed to multiple recipients  
25 -- further consuming bandwidth resources.

In an effort to overcome bandwidth constraints, data and geometry compression has previously been used to reduce information content in 2D and 3D models. Previous compression attempts include image compression (e.g., JPEG), defining objects with shared features (e.g., shared edges), small texture maps for large areas, etc. Examples of some of these and other techniques can be found in U.S. Patent No. 5,740,409 which teaches a 3D graphics accelerator for compressed geometry, and U.S. Patent Nos. 5,793,371, 5,867,167, and 5,870,094 which teach various methods for more-efficiently encoding 3D models.

In addition to geometry compression, general purpose data compression procedures have also been applied to model data. Such techniques include *Huffman* encoding (See Huffman, "A Method For Construction Of Minimum Redundancy Codes", Proceedings IRE, 40, 10 pages 1098-1100 (Sep. 1952)), *Tunstall* encoding (See Tunstall Doctoral thesis, "Synthesis of Noiseless Compression Codes", Georgia Institute of Technology (Sept. 1967)), and *Lempel-Ziv* encoding (See "A Universal Algorithm For Sequential Data Compression", IEEE Transactions on Information Theory, IT-23, 3, pages 337-343 (May, 1977)), and *run-length* encoding of model data (see, e.g., U.S. Patent No. 3,656,178).

Even though application of general purpose and geometric compression has increased the viability of multiple-player games and transmission of a large volume of geometry data over a network connection, there still remains a significant amount of information that needs to be transferred before such games are comparable to traditional games having access to locally stored model data. The situation applies equally to other applications involving multi-media content streaming, e.g. video conferencing.

What is needed then, is some way to further improve the way model data are streamed to network players, conferencing applications, and the like.

## SUMMARY OF THE INVENTION

A novel method for streaming multi-media content is disclosed. Multiple  
5 versions of model data tailored for different operating environments differentiated in  
accordance with value(s) of at least one operating characteristic of remote  
requesting client computer systems are stored in a multi-media content providing  
server. A multi-media content player of a client computer system determines the  
operating characteristic value(s) for the at least one operating characteristic of the  
10 client computer system. The multi-media content player adaptively requests  
appropriate versions of selected ones of the model data, based at least in part on  
the determined operating characteristic value(s) of the at least one operating  
characteristic of the client computer system. In response, the providing server  
streams the requested versions of the requested model data to the multi-media  
15 content player for rendering.

## BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references  
5 denote similar elements, and in which:

**Figure 1** illustrates an overview of the dynamic scalable content streaming method and apparatus of present invention, in accordance with one embodiment;

**Figure 2** illustrates an example table for assigning different weights to different operating characteristics employed to perform the adaptive request of  
10 model data of the present invention, in accordance with one embodiment;

**Figure 3** illustrates an example content of a multi-media content title, in accordance with one embodiment;

**Figure 4** illustrates an example organization of model data on a multi-media content providing server, in accordance with one embodiment;

**Figure 5** illustrates a block diagram view of the multi-media content player of the present invention, in accordance with one embodiment;

**Figure 6** illustrates the operational flow of the end-user interface of the multi-media content player of the present invention, in accordance with one embodiment;

**Figure 7** illustrates the operational flow of the “new title” script of the multi-media content player of the present invention, in accordance with one embodiment;

**Figure 8** illustrates the operational flow of an example script of a multi-media content title of the present invention, in accordance with one embodiment; and

**Figure 9** illustrates an example computer system suitable for practicing as either a content creation system or a content consumption system, in accordance  
25 with one embodiment.

## DETAILED DESCRIPTION OF THE INVENTION

In the following description, various aspects of the present invention will be described, and various details will be set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all aspects of the present invention, and the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

Parts of the description will be presented using terminology commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art, such as multi-media contents, video, audio, graphics, texts, and so forth. Also, parts of the description will also be presented in terms of operations performed by a computer system, using terms such as, determining, requesting, receiving, rendering, and the like. As well understood by those skilled in the art, these quantities or operations take the form of electrical, magnetic, or optical signals being stored, transferred, combined, and otherwise manipulated through electrical, magnetic and optical components of a digital system; and the term digital system includes general purpose as well as special purpose data processing machines, systems, and the like, that are standalone, adjunct or embedded.

Various operations will be described as multiple discrete steps performed in turn in a manner that is most helpful in understanding the present invention. However, the order of description should not be construed as to imply that these operations are necessarily performed in the order they are presented, or even order dependent. Lastly, repeated usage of the phrase "in one embodiment" does not necessarily refer to the same embodiment, although it may.

Referring now **Figure 1**, wherein a block diagram illustrating an overview of the present invention in accordance with one embodiment is shown. As illustrated, in accordance with the present invention, multi-media content providing server **102** stores multiple versions of compartmentalized model data **106** (of multi-media contents), tailored for different operating environments differentiated in accordance with value(s) of at least one operating characteristic of multi-media content consuming client computer systems **104**. Server **102** is communicatively coupled to client computer systems **104** via communication links **103**. Each client computer system **104** is provided with a multi-media content player **108** incorporated with the teachings of the present invention. More specifically, the operating characteristic value(s) for the at least one operating characteristic of a client computer system **104** is (are) determined. The multi-media content player **108** adaptively requests appropriate versions of selected ones of model data **106**, based at least in part on the determined operating characteristic value(s) of the at least one operating characteristic of the client computer system **104**. In one embodiment, the multi-media content player **108** further adapts the requests in accordance with one or more monitored performance indicators of the client computer system **104**. In response, server **102** streams the requested versions of the requested model data **106** to multi-media content player **108** for rendering. As a result of the adaptive request and streaming, user experience at the various client computer systems **104** are improved.

Except for the manner model data **106** are organized to enable client computer systems **104** to adaptively request streaming of different versions of selected ones of the model data in accordance with its operating environment (and optionally, its operating condition), server computer system **102** is intended to



represent a broad range of communication enabled servers known in the art. Examples of suitable server to practice the present invention include Netfinity Servers, available from IBM of Armonk, New York, and Enterprise Servers, available from Sun Microsystems of Menlo Park, CA. Similarly, except for the incorporation of  
5 player **108** of the present invention, client computer systems **104** are intended to represent a broad range of communication enabled personal computer systems, personal digital assistants (PDA), set-top boxes and "electronic appliances" of the like, known in the art. Examples of suitable client computer systems to practice the present invention include Dimension Computers from Dell Computer of Austin, TX,  
10 iMAC from Apple Computer of Cupertino, CA, and Palm VII from 3COM of San Jose, CA.

Communication links **103** are intended to represent a broad range of wired and wireless communication medium known in the art. Examples of appropriate wired communication links include direct modem connections over conventional  
15 Public Switching Telephone Network (PSTN) lines, private data network connections through e.g. Frame Relay or asynchronous transfer mode (ATM) networks, and public data network connections through e.g. the Internet. Examples of appropriate wireless communication links include cellular as well as satellite connections.

Multi-media contents of which model data **106** are a part is intended to  
20 represent a broad range of entertainment, educational, business and other multi-media contents of the like. In one embodiment, multi-media contents are multi-media games, where compartmentalized model data **106** include but not limited to geometry data, lighting data, coloring data, texturing data, animation data, and audio data associated with various models, e.g. a car, a person, an animal and so forth.  
25 The various versions of model data **106** are model data **106** compressed with various lossy compression techniques to provide various tradeoff levels between

precision and transmission bandwidth requirement. These lossy compression techniques include but not limited to those known in the art, as well as compression techniques disclosed in co-pending U.S. patent application, number <to be assigned>, contemporaneously filed, and entitled "Data Compression Through Adaptive Bit Size Reduction", and co-pending U.S. patent application, number <to be assigned>, contemporaneously filed, and entitled "Data Compression Through Offset Representation",, which are hereby fully incorporated by reference.

In one embodiment, the at least one operating characteristic includes the bandwidth available on communication link **103**, i.e. the amount of data bits that can be transferred over link **103**, the power of the processor included in a client computer system **104**, e.g. in terms of its model, operating speed and/or MIP rating, the amount of system memory and swap space available in a client computer system **104**, e.g. in terms of Mbytes, the operating speed of its memory and system bus, e.g. in terms of MHz, the availability of video memory, also in Mbytes, whether the processor is augmented with digital signal processing for audio decompression and hardware acceleration for graphics. In other embodiments, more or less operating characteristics may be employed.

In one embodiment, the values indicative of these operating characteristics (and others) are determined at least at installation time of player **108** on a client computer system **104**. That is, player **108** includes an auxiliary operating characteristic determination function, and the installation utility responsible for installing player **108** on a client computer system **104** invokes this auxiliary operating characteristic determination function to perform the determinations in accordance with the teachings of the present invention as an integral part of the installation operation. Installation and the manner the values indicative of these and other operating characteristics may be determined are operating environment dependent.

For the Window® family of operating systems (available from Microsoft Corporation of Redmond, WA), installation may be accomplished via the well known “set-up” approach, and the values for these operating characteristics may be obtained by querying the registry of the operating system, and if necessary, the various

5 configuration files, i.e. config.sys, system.ini, window.ini, and so forth. For Intel® architecture compatible system, if necessary, the values may also be obtained by accessing the configuration information maintained by the Basic Input/Output System (BIOS). In other environments, if necessary, some of these values may be determined through benchmarking. For example, in the case of communication  
10 bandwidth, it may be determined by performing a number of sample transmissions over a period of time. In the case of memory size, it may be determined by making successive accesses to larger and larger addresses (starting from a low address) or to lower and lower addresses (starting from an arbitrary large address).

In one embodiment, where values for multiple environmental characteristics  
15 are determined, a composite operating environment index (COEI) is formed based on the determined values to allow all the environmental characteristics to be considered simultaneously. In one embodiment, the COEI is formed by assigning different weights to the determined values of the different environmental characteristics. In one embodiment, the values for each environmental character is  
20 normalized into a range where the most capable end of the range is assigned the weight of 1, the least capable end of the range is assigned the weight of 0, and the intermediate capabilities are assigned corresponding values between 0 and 1 (see the example weight assignment table of **Fig. 2**). In alternate embodiments, other weighting factors, as well as other approaches may be employed to facilitate the  
25 simultaneous considerations of multiple operating environmental characteristics.

In one embodiment, the performance indicators of client computer systems **104** monitored include bandwidth utilization, CPU utilization, memory utilization, memory swapping, cache hit rate, and audio frames drop rate. In other embodiments, more or less performance indicators may be monitored. In one

5 embodiment, the current values for these performance indicators are obtained by an auxiliary performance monitor provided to player **108**, making periodic queries to the operating system. In other embodiments, one or more of these performance indicators (when not supported by the operating system) may be benchmarked by the auxiliary performance monitor itself.

10 In one embodiment, where the above described COEI is employed to facilitate simultaneous consideration of multiple operating environment characteristics, the auxiliary performance monitor, in response to the receipt of the performance values from the underlying operating system, temporarily adjusts the COEI up or down to influence the selected ones, and the versions of the selected

15 ones of model data **106** adaptively requested by player **108**.

**Figure 3** illustrates an example organization of a multi-media content title, in accordance with one embodiment. As illustrated, multi-media content title **300** is an interactive multi-media content title. It includes one or more scripts **302** specifying

20 various actions to be taken at initialization time of the title, in response to user inputs, and when user inputs are not present. In addition to scripts **302**, title **300** includes model data **106**. For the illustrated embodiment, model data **106** includes a number of background scene objects **304**, foreground model objects **306** and audio objects **308**. At least the foreground model objects **306** are

25 compartmentalized into the earlier described geometry data, texture data, coloring data, lighting data, animation data and so forth. In other embodiments, more planes

(i.e. foreground and background) as well as more or less compartmentalization (i.e. geometry, color, and so forth) may be employed.

**Figure 4** illustrates the organization of model data **106** in accordance with one embodiment, where the above described COEI is employed. For the illustrated embodiment, for at least each of the geometry data objects,  $n$  versions of the data object representing different tradeoff levels between precision and transmission bandwidth requirements are maintained. For the sample geometry data object A, its version-0 corresponds to an uncompressed version, to be used in operating environments with a COEI score range of  $(x_1, \text{max})$ , where  $x_1$  is smaller than max; its version-1 corresponds to a compressed version, to be used in operating environments with a COEI score range of  $(x_2, x_1)$ , where  $x_2$  is smaller than  $x_1$ ; and so forth. For different data objects,  $n$  as well as the COEI score ranges may be different.

**Figure 5** illustrates player **108** in further detail in accordance with one embodiment. As illustrated, player **108** includes end-user interface (EUI) **502**, script execution engine **504**, and "new title" script **506**. End-user interface **502** is responsible for facilitating interaction with a user of player **108**. Script execution engine **504** is responsible for executing "new title" script **506** and scripts of multimedia content titles (see **Fig. 3**). Additionally, as described earlier, player **108** further includes first and second auxiliary functions **508** and **510**. First auxiliary function **508** is invoked to determine the values of the various operating characteristics, and COEI, if applicable, as described earlier. Second auxiliary function **510** is invoked to monitor the various performance indicators of interest of client computer systems **104**, as described earlier.

Script execution engine **504** is intended to represent a broad range of “script” execution environment known in the art. In one embodiment where script **506** and scripts of multi-media content titles are implemented using Java™Script or script languages of the like, script execution engine **504** is a Java™Script execution environment or a script execution environment of like kind.

EUI **502**, script **506**, and an example script of a multi-media content title are described below referencing **Figures 6-8**.

Referring now to **Figure 6**, wherein a block diagram illustrating the operational flow of EUI **502** in accordance with one embodiment is shown. As shown, at **602**, EUI **502** awaits inputs from a user. Upon receipt of an user input, EUI **502** determines if the input is for a current executing script of a multi-media content title, **604**. If so, EUI **502** forwards the user input to the current executing script for handling, **606**. Upon doing so, EUI **502** continues operation at **602**. On the other hand, if the input is not intended for a current executing script, EUI **502** further determines if the user is attempting to start a new title, **608**. If so, EUI **502** starts execution of the earlier described “new title” script, **610**. In like manner, upon doing so, EUI **502** continues operation at **602**. However, if the user input is not intended to start a new title, EUI **502** handles the input (e.g. an user command) in an application dependent manner.

**Figure 7** illustrates the operational flow of “new title” script **506** in accordance with one embodiment. As illustrated, at **702**, “new title” script **506** determines if the requested title has previously been downloaded. If the requested title has not been previously downloaded, “new title” script **506** requests downloading of the new title from server **102** accordingly, **704**; otherwise, “new title” script **506** skips request

operation **704**. If request of the new title is necessary, “new title” script **506** may additionally request the operating system to establish a communication connection to server **102**, if a connection to server **102** does not currently exist. In any event, upon determining that the requested title was previously downloaded, or upon  
5 receiving the newly requested title, “new title” script **506** causes execution of the (initial or main) script of the requested title to begin, **708**.

For the illustrated embodiment, upon causing execution of the (initial or main) script of the requested title to begin, “new title” script **506** further determines if the auxiliary performance monitor function **510** have been launched, **710**. If the  
10 auxiliary performance monitor function **510** have been launched, no further action are taken, otherwise, “new title” script **506** further causes the auxiliary performance monitor function **510** to be launched, **712**, to monitor various performance indicators of interest of client computer system **104**, and to adjust COEI, as described earlier, to influence the selected ones and the versions of the selected ones of model data  
15 **106** being requested.

**Figure 8** illustrates the operational flow of an example script of a multi-media content title in accordance with one embodiment. As illustrated, at **802**, the example script requests an initial round of foreground and background data objects. In  
20 accordance with the present invention, the script is designed to request a small set of data objects, to reduce a user’s perception of start-up time. In addition to limiting the initial set of data objects to a small set, the script further requests versions of the data objects that tradeoff precision for reducing transmission bandwidth requirement, consistent with the COEI of the client computer system, to further  
25 reduce a user’s perception of start-up time. In one embodiment, the request is made by calling appropriate ones of the communication services of the operating

system to package and deliver the request to server **102**. For the illustrated embodiment, the script further requests the coloring, lighting, texture and animation objects, after requesting the geometry and audio objects first, to enhance the likelihood that the geometry and audio objects will be returned prior to the secondary  
5 objects, that is coloring, lighting, animation and so forth, to again further reduce the user's perceive start-up time. In an alternate embodiment, the script may even forego requesting some of the associated data objects, i.e. lighting, texture, and so forth. At **804-806**, the script renders the requested model data as they arrive, making the necessary calls to the operating system's graphics and audio services.  
10 For the illustrated embodiment, the script will automatically synchronize rendering for at least the audio object in view of the timing of its arrival. That is, if for example, the audio object is for 1 min. of audio, and execution of the title has proceeded to .25 min beyond when the audio object was needed, the script will skip or drop the first .25 min of audio, and commence with the rendering at .26 min into the audio  
15 object. As a result of requesting a small initial set of objects, the order of requesting these objects, the transmission friendly manner of requesting (i.e. the low bandwidth requirement versions, as well as the optional foregoing of some of the associated data objects), the script is able to give the user the perception of earlier operation than otherwise possible.

20 At **808**, the script awaits the user's inputs. For the illustrated embodiment, while there are no inputs, the script determines if some of the downloaded objects should be upgraded to include previously forego associated data objects, or to higher precision versions of these data objects, **810**. The determination is made based at least in part on the monitored performance indicators, by examining the  
25 COEI to determine if it has been updated to a higher performance level. If so, the



script requests the updates, i.e. the previous foregoing data objects and/or the higher precision versions of some of the existing objects, **812**.

If user inputs were received at **808**, the script determines if any of the objects are to be re-rendered in a modified manner, e.g. at a different location or with a different animation version, and whether new objects are to be rendered, **814**. If not, the script proceeds to **810** and continues its operation as described earlier. If either at least one of the object is to be re-rendered in a modified manner, or a new object is to be rendered, the script determines if it has all the data objects necessary to accomplish the task, **816**. If additional data objects are required, the script requests versions of the required data objects consistent with the operating environment and the current operating condition, **818**. For the illustrated embodiment, as described earlier, the appropriate version is determined based at least in part on the current performance level indicated by COEI. In any event, upon determining that it has the necessary data objects or upon receipt of the newly requested data objects, the script causes the data objects to be rendered, making the necessary calls to the operating system's graphics and audio services, as described earlier, **820**. Upon rendering the data objects, the script proceeds to **810** and continues as described earlier.

**Figure 9** illustrates one embodiment of a computer system suitable for use to practice as either a server system or a client system of the present invention. As shown, computer system **900** includes processor **902** and memory **904** coupled to each other via system bus **906**. Typically (although not necessary), for use as a server, i.e. for storage and provision of model data **106** of multi-media content titles, processor **902**, memory **904**, and system bus **906** are usually higher performance models of these elements, whereas for use as a client system, i.e. execution of

player **108**, the lower performance models are employed instead. Coupled to system bus **906** are non-volatile mass storage **908**, such as hard disks, floppy disk, and so forth, input/output devices **910**, such as keyboard, displays, and so forth, and communication interfaces **912**, such as modem, LAN interfaces, and so forth. Each of these elements perform its conventional functions known in the art. In particular, system memory **904** and non-volatile mass storage **908** are employed to store a working copy and a permanent copy of the programming instructions implementing the above described teachings of the present invention, e.g. an underlying operating system, and model data **106** or player **108**. As described earlier, the permanent copy of the programming instructions implementing model data **106** or player **108** is preferably loaded through communication interfaces **912**, although it may be loaded through fix form distribution medium, such as diskettes, CD or DVD. The constitution of elements **902-914** are well known, and accordingly will not be further described.

In general, those skilled in the art will recognize that the present invention is not limited by the details described, instead, the present invention can be practiced with modifications and alterations within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of restrictive on the present invention.

Thus, a method and apparatus for dynamic scalable content streaming has been described.